REMARKS

Claims 1-29 are pending in the present application. Claims 1, 4, 5, 9, 11-13, 15, 16, 22, 25, 28 and 29 have been amended. No claims have been added or canceled. Applicant respectfully requests reconsideration and allowance of the application.

<u>35 U.S.C. § 103 Rejections</u>

<u>Claims 1-7, 10-17 and 19-29</u>

Claims 1-7, 10-17 and 19-29 stand rejected under 35 U.S.C. 103(a) as being anticipated by U.S. Patent Number 5,937,331 issued to Kalluri, et al (hereinafter "Kalluri") and U.S. Patent Number 5,689,501 issued to Takase, et al. (hereinafter "Takase") and U.S. Patent No. 5,802,582 issued to Chess, et al. (hereinafter "Chess"). Applicant respectfully traverses the rejection.

Kalluri discloses a protocol and system for transmitting triggers from a remote network and for controlling interactive program content at a broadcast station. Although Kalluri provides a redundancy mechanism for recovery from non-received or corrupted trigger commands, Kalluri does not disclose or suggest connecting to the server system to receive replacement instructions.

Claim 1, as amended, recites:

"In a client system that communicates with a server system, wherein the client system includes a computing device that includes <u>stored</u> program instructions used in the operation of the client system, a method of restoring a corrupted portion of the <u>stored</u> program instructions at the client system...."

The method includes steps of: (1) "checking the validity of <u>stored</u> system program instructions at the client system and checking the validity of <u>stored</u> application program instructions at the client system to determine whether the <u>stored</u> system program instructions or the <u>stored</u> application program instructions have a corrupted portion so as to render the corrupted portion unreadable as intended;" and (2) "upon determining that either the <u>stored</u> system instructions or the <u>stored</u> application program instructions have a corrupted portion," performing steps of: (a) connecting the client system to the server system;" (b) "receiving replacement instructions for the corrupted portion from the server system;" and (c) "replacing the corrupted portion of the stored program instructions with the replacement instructions."

Claim 1 describes an active system that takes steps to replace corrupted instructions upon discovering that such corrupted instructions exist. By contrast, Kalluri describes a passive system that is continuously connected to a server and checks the integrity of transmissions. When a corrupted portion of a transmission is detected (by the absence or mis-ordering of particular triggers) the corrupted portion is re-transmitted.

Takase describes a process for Automated Teller Machines (ATM) communications. Communication packets are checked for accuracy during the communication process and corrupted packets are handled as an exception.

Chess describes an automated update system for alterable memory in a computer, such as a BIOS stored in FLASH ROM and the like. The integrity of the BIOS is checked periodically and updates to the BIOS are not performed if the existing BIOS is corrupted.

The amendments to claim 1 make it clear that the corrupted instructions are stored on the client system. This is different from the disclosures of Kalluri and Takase since those disclosures deal with verifying communication transmissions. Although Chess describes programs stored in memory, Chess does not describe any method to correct corruptions found in such stored programs.

Another difference that claim 1 shows over the cited reference is the fact that claim 1 requires that the client system establish a connection with a server in the event of determining that a file corruption exists. Neither Kalluri nor Takase contemplates such an action because the integrity verification described therein occurs after a connection has been established and, in fact, while the client is connected to the server.

Chess does not make any mention whatsoever of connecting to a server in the event of the identification of corrupted program instructions.

The cited references do not teach or suggest the elements recited in claim 1. Taken as a whole, claim 1 describes finding corrupted instructions in a program stored on a client computer and thereafter establishing a connection with a server computer so that un-corrupted instructions can be downloaded to replace the corrupted instructions. There is simply no combination of the references that can be made to stand for this functionality.

Accordingly, claim 1 is allowable over the cited references and the rejection of claim 1 should therefore be withdrawn.

Claims 2-7, 10 and 11 depend from claim 1 and are allowable at least by virtue of that dependency as well as for other reasons. Therefore, the rejection of these claims should also be withdrawn.

Claim 12 has been amended and now recites "A computer program product for implementing, in a client system that communicates with a server system, wherein the client system includes a computing device that includes stored program instructions used in the operation of the client system...." The computer program product includes "a computer-readable medium carrying computer-executable instructions, that when executed at the client system, cause the client system to perform the method...." The method includes acts of: (1) "checking the validity of stored system program instructions at the client system and checking the validity of stored application program instructions at the client system to determine whether the stored system program instructions or the stored application program instructions have a corrupted portion so as to render the corrupted portion unreadable as intended;" and (2) "after determining that either the stored system instructions or the stored application instructions have a corrupted portion:

connecting the client system to the server system;

receiving replacement instructions for the corrupted portion from the server system; and

replacing the corrupted portion of the stored program instructions with the replacement instructions."

Similar to claim 1, claim 12 requires determining if stored instructions have been corrupted and, in the event a corruption has been found, connecting to a server to download replacement instructions. None of the references nor a combination thereof teach or suggest the elements recited in claim 12.

By the same rationale stated in the response to the rejection of claim 1, claim 12 is allowable over the cited references. Therefore, the rejection of claim 12 should be withdrawn.

Claims 13-15 depend from claim 12 and are allowable at least by virtue of that dependency as well as for other reasons. Therefore, the rejection of these claims should also be withdrawn.

Claim 16, as amended, recites:

"In a client system that communicates with a server system, wherein the client system includes a computing device that includes stored program instructions used in the operation of the client system, a method of restoring a corrupted block of the stored program instructions at the client system, comprising the acts of:

checking the validity of blocks of the stored program instructions at the client system to determine whether the blocks of the stored program instructions include a corrupted block so as to render the corrupted block unreadable as intended; and

upon determining that the blocks of the program instructions include a corrupted block:

requesting the replacement block of program instructions for the corrupted block from the server system;

receiving the replacement block of program instructions from the server system; and

replacing the corrupted block with the replacement block of program instructions."

Claim 16 contains similar limitations as claim 1 and claim 12, namely, that stored instructions are checked for corruption and, if found, a subsequent connection is made to a server to find replacement instructions for the corrupted instructions.

For the same reasons as discussed in the response to the rejection of claims 1 and 12, claim 16 is allowable over the cited references.

Claims 17 and 19-24 depend from claim 16 and are allowable at least by virtue of that dependency as well as for other reasons. Therefore, the rejection of these claims should also be withdrawn.

Claim 25 has been amended to recite "A computer program product for implementing, in a client system that communicates with a server system, wherein the client system includes a computing device that includes stored program instructions used in the operation of the client system, a method of restoring a corrupted block of the stored program instructions at the client system, the computer program product comprising:

a computer-readable medium carrying computer-executable instructions that when executed at the client system, cause the client system to perform the method, including the acts of:

checking the validity of blocks of the stored program instructions at the client system to determine whether the blocks of the stored program instructions include a corrupted block so as to render the corrupted block unreadable as intended; and after determining that the blocks of the <u>stored</u> program instructions include a corrupted block:

requesting the replacement block of program instructions for the corrupted block from the server system;

receiving the replacement block of program instructions from the server system; and

replacing the corrupted block with the replacement block of program instructions."

Claim 25, similar to claims 1, 12 and 16, include elements of checking for corrupted instructions that are stored on a client device and, if such corrupted instructions are found, connecting to a server to download replacement instructions.

By the same rationale presented in the responses to the rejections of claims 1, 12 and 16, claim 25 is allowable over the cited references. Accordingly, the rejection of claim 25 should be withdrawn.

Claims 26-29 depend from claim 25 and are allowable at least by virtue of that dependency as well as for other reasons. Therefore, the rejection of these claims should also be withdrawn.

<u>Claims 8, 9 and 18</u>

Claims 8, 9 and 18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Kalluri, Takase, Chess, and U.S. Patent No. 5,903,732 issued to Reed, et al. (hereinafter "Reed"). Applicant respectfully traverses the rejection.

Claims 8 and 9 depend from claim 1 and are allowable at least by virtue of that dependency. **Claim 18** depends from claim 16 and is allowable at least by virtue that dependency.

Reed does not cure the deficiencies of the other references as point out above. Therefore, claim 8, 9 and 18 are allowable over the cited references and the rejection to these claims should also be withdrawn.

CONCLUSION

Accordingly, in view of the above amendment and remarks it is submitted that the claims are patentably distinct over the cited references and that all the rejections to the claims have been overcome. Reconsideration of the above Application is requested. Based on the foregoing, Applicants respectfully requests that the pending claims be allowed, and that a timely Notice of Allowance be issued in this case. If the Examiner believes, after this response, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicants hereby request any necessary extension of time. If there is a fee occasioned by this response, including an extension fee that is not covered by an enclosed check please charge any deficiency to Deposit Account No. 50-0463.

Respectfully submitted,
Microsoft Corporation

Date: October 3, 2006

By: ______________________

James R. Banowsky, Reg. No.: 37,773
Attorney for Applicant
Direct telephone (425) 705-3539
Microsoft Corporation
One Microsoft Way
Redmond WA 98052-6399

CERTIFICATE OF MAILING OR TRANSMISSION
(Under 37 CFR § 1.8(a)) or ELECTRONIC FILING

I hereby certify that this correspondence is being electronically deposited with the USPTO via EFS-Web on the date shown below:

October 3, 2006
Date

Noemi Tovar